

POLY 88

Users Group

Software Library

Title _____	Game Tape #2
Hardware Requirements _____	16K RAM
Software Requirements _____	BASIC Version A00
By _____	

PolyMorphic
Systems

Game Tape #2

Features Four Games:

BACKGAMMON, a video version of the traditional game.

HANGMAN, stake your life against your word guessing skill.

WUMPUS, hunt a monster never seen by human eyes. Watch out for pitfalls and superbats.

CRAPS, roll the dice and take your chances!

To use these programs, first load BASIC version A00. Make sure this tape is rewound. Now, ZERO the counter on your cassette recorder. Use the counter to find the approximate starting position for the program you intend to use.

<u>Program</u>		<u>Counter Position</u>
BACK (BACKGAMMON)	0-18	000
HANGMAN	0-13	055
WUMPUS	0-19	102
CRAPS	0-19	177

The counter position is only an approximation and is based on a Super-scope C103-A. Only one program may be loaded at a time. Brief descriptions and listings for each program follow this page.

PROGRAM NAME: BACKGAMMON

TYPE: GAME

HARDWARE REQUIREMENTS: 16K RAM

SOFTWARE REQUIREMENTS: BASIC VERSION A00

DESCRIPTION:

- Point: one of the 24 positions on the board. These are represented as elongated triangles on a classic Backgammon Board. In this computer version, each point is represented by a slot between vertical stripes. The points are labeled A through X.
- Man: a playing piece represented by a "U" or "arch" shaped graphic character.
- Home Court: the last six points that a given player can move to. The first player's home court consists of the lower right 6 points. The second player's home court consists of the upper right 6 points.
- Bar: a type of prison for men who have been hit. With the classic board, the center stripe (which bisects the board) is the bar. This computer game uses the area adjacent to the player's name (at the right).
- Player: one of the people playing the game.
- Bear Off: to remove your own men from the card.
- Board: the playing board consisting of 24 points.
- Pyramid: a group of adjacent points each held by two or more of one player's men. This forms a block or partial block to the opponent's movement.
- Hit: to move to a point occupied by only one of the opponent's men. The opponent's man is then removed from the board and placed on the bar.

BACKGAMMON

Blot: a single player on a point.

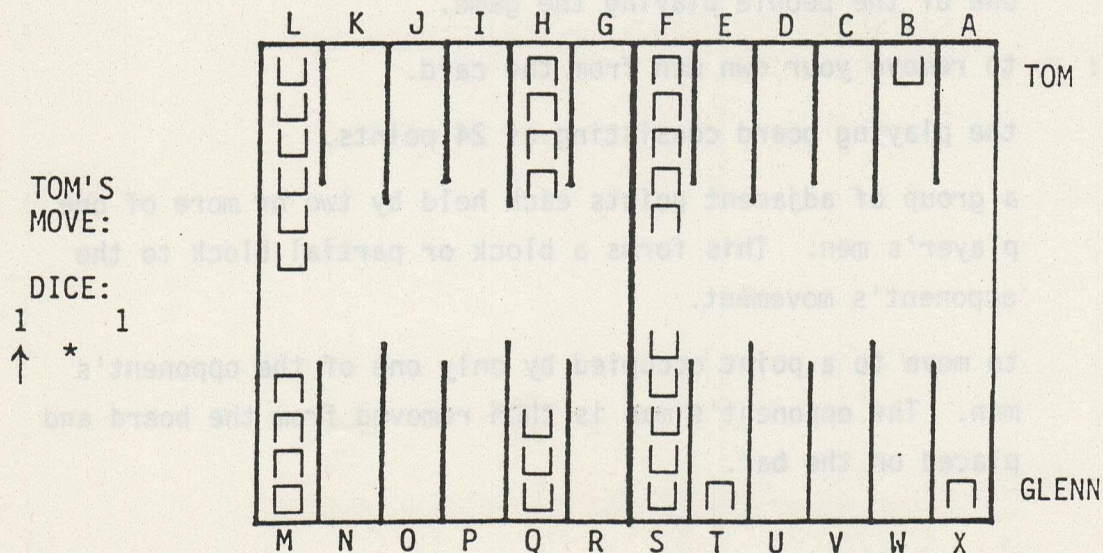
The object of Backgammon is to bear off all of your men.

There are two players and fifteen men per player. The first player moves his men counter-clockwise around the board. The second player moves clockwise.

The men are moved according to the throw of the dice (the computer rolls the dice in this game). Unlike most other dice games, backgammon treats each die separately. You must move a man forward the number of points indicated by one die, if possible. Then you move any man (including the one just moved) according to the second die. You cannot move to a point occupied by two or more of your opponents men. Example 1: if your dice are 4 and 5, you can move one man four points and a second man five points or you can move a single man 9 points if, and only if, he could have moved to an intermediate position (the fourth or fifth point in this example).

Doubles (both dice have the same number) constitute a special case; each die is moved twice. Doubles are indicated by an asterisk below the dice values.

Fig.1.



BACKGAMMON

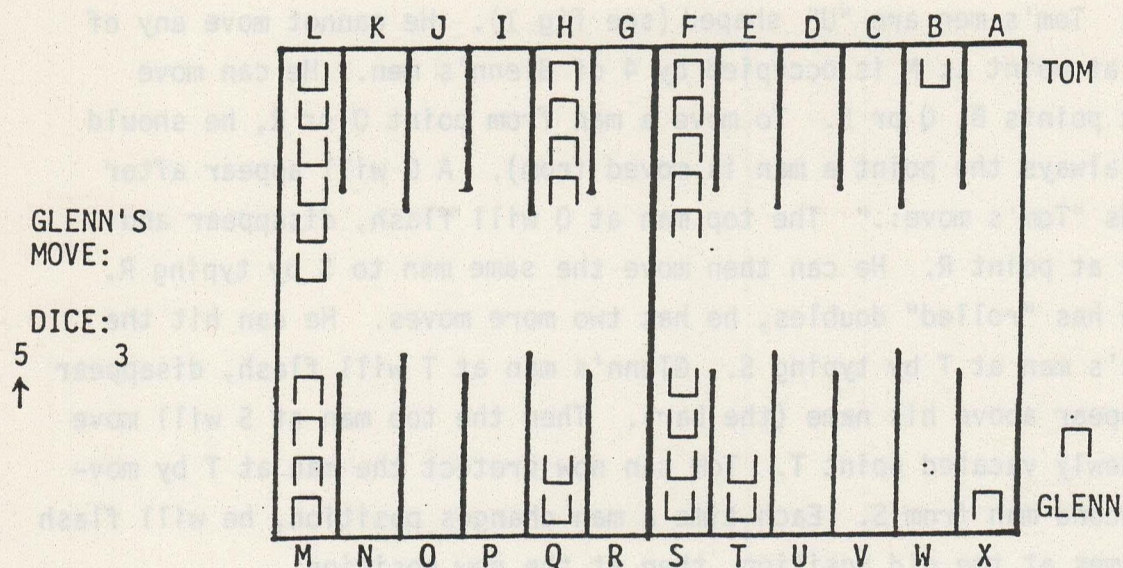
Example: Tom's men are "U" shaped (see fig 1). He cannot move any of the men at point L; M is occupied by 4 of Glenn's men. He can move those at points B, Q or S. To move a man from point Q or R, he should type Q (always the point a man is moved from). A Q will appear after the words "Tom's move:." The top man at Q will flash, disappear and reappear at point R. He can then move the same man to S by typing R. Since he has "rolled" doubles, he has two more moves. He can hit the opponent's man at T by typing S. Glenn's man at T will flash, disappear and reappear above his name (the bar). Then the top man at S will move to the newly vacated point T. Tom can now protect the man at T by moving a second man from S. Each time a man changes position, he will flash a few times at the old position, then at the new position.

The computer will roll the dice after a player finishes his moves. The arrow points at the die the computer will work with first. If the player wishes to move the other die first, he can type a space (space bar). This command will switch the dice positions. The arrow will move to the second die after the player's first move. All of a player's move can be typed at once if the player can anticipate the intermediate positions. Caution is important when using the "type-ahead" feature; a move cannot be retracted once the computer begins moving a man. The control X command (hold down the CTRL key while typing an X) can be used to clear commands typed ahead, but only those not being implemented by the computer.

All of a player's men must be returned from the bar before the player can move other men. A man is returned by typing the return (carriage return) key. The man is returned by placing him on one of the points next to the player's name according to the die value. The starting point is A for the first player and X for the second player.

BACKGAMMON

Fig. 2



Example 3: Figure 2 shows the final positions for example 2 with new dice values. Glenn must return his man from the bar before moving any other man. Glenn cannot return on a 5 as Tom has two men at point T. He must type a space followed by a return. This moves the man on the bar to point V. Glenn can then move one of the players at point M, H or F (other moves are blocked). Glenn chooses to move a man from M to H so he types an M.

If a player has no possible move for a die, he can forfeit the move by typing the escape key (this is a control "[" on many keyboards, - hold down the CTRL and SHIFT keys while typing K). Backgammon rules require a player to move if possible, however, the computer will not prevent a player from forfeiting any move. Thus, you should check with your opponent and agree you have no possible move before forfeiting the move. If moving one die makes the second die useable, both dice must be moved.

When all of a player's men are in his home court, he can begin bearing off. This is accomplished by moving men past the last position on the board. The die can have any value greater than the number of points in front of the man bearing off. Given the choice of bearing off or moving a man forward (without bearing off), a player must move a man

BACKGAMMON

forward. The men furthest back must be moved first. This is the second type of illegal move the computer will not prevent.

The first player to bear off all his men wins. Note, it is possible to be hit while bearing off. You must stop bearing off until the man hit is moved back into home court.

The computer will prevent a number of improper moves. The following is a list of error messages and their meanings:

1. Other player! - You have tried to move your opponent's man.
2. Can't bear off - you do not have all your men in your home court. You cannot bear off until they are.
3. Man on bar! - You must return the man from the bar before moving other men.
4. Can't hit! - You cannot hit your opponent if he has 2 or more men on the point.
5. Point empty - there is no man on the point you are trying to move from.
6. No man on bar - you have hit the return key, but there is no man to be returned.

Strategies:

1. Avoid blots, when possible as they are susceptible to being hit. This is especially true near or in home court.
2. Build pyramids - pyramids of six or more adjacent points prevent the opponent from passing. Pyramids of 5 or less adjacent points restrict the opponent.
3. Sometimes leaving a single man in your opponent's home court proves handy. It is almost inevitable that he will create a blot while bearing off.

Many books have been written on the game of Backgammon. See your local library or bookstore for more information, rules and strategies.


```

LIST 0,650
100 REM: *** BACKGAMMON 1.0 ***
110 REM: AUTHOR - GLENN A. MCCOMB (APRIL 1977)
120 DIMB(26),R(16),M$(32),Z$(32)
130 Z=RND(TIME(0)/65535)\!CHR$(12)
140 A$=FNI$("PLAYER A?",5,28,8)\B$=FNI$("PLAYER B?",5,28,8)
150 GOSUB 750\GOSUB 720
160 B(25)=0\B(26)=0\GOSUB 650
170 Z=FND(57,13,B$)\ Z=FND(57,2,A$)
180 Z=FND(1,6,"DICE:")
190 P=1\W$=A$
200 H=0\GOSUB 580
210 GOSUB 1720
220 FOR M=1 TO 2
230 D=D1\IF M=2 THEN D=D2
240 F=FNG(M)
250 IF F<1 THEN ON F+3 GOTO 350,330,400
260 Z1=(P*P+P)/2\Z2=ABS(B(Z1+25))\IF Z2>0 THEN 460
270 IF B(F)=0 THEN 480 ELSE IF SGN(P)<>SGN(B(F)) THEN 440
280 T=F+D*P
290 IF T<1 OR T>24 THEN T=0\ IF FNB(P)=0 THEN 450 ELSE 380
300 IF P=SGN(B(T)) OR B(T)=0 THEN 380
310 IF ABS(B(T))>1 THEN 470
320 Z=FNM(T,0)\Z=FNH(-P,1)\GOTO 380
330 F=((P*P+P)/2)+25\IF B(F)=0 THEN 490
340 Z=(P*P-P)/2\T=ABS(Z*25-D)\GOTO 300
350 IF H=1 THEN 230
360 D=D1\D1=D2\D2=D\Z=FNA(1,7,D1+176)\Z=FNA(5,7,D2+176)
370 GOTO 230
380 IF F>24 THEN Z=FNH(P,-1)
390 Z=FNM(F,T)\H=1\Z=FNW(P)
400 NEXT
410 IF D3=1 THEN D3=0\GOTO 220
420 IF P<0 THEN 190 ELSE P=-1\W$=B$
430 GOTO 200
440 M$="Other player!"\ GOTO 500
450 M$="Can't bear off..."\ GOTO 500
460 M$="Man on bar!"\ GOTO 500
470 M$="Can't hit..."\ GOTO 500
480 M$="Point empty..."\ GOTO 500
490 M$="No man on bar..." \ GOTO 500
500 J$=""\L=0\N=1\K=LEN(M$)
510 IF N>K THEN 540 ELSE K$=M$(N,N)\N=N+1
520 IF K$="" THEN 540
530 J$=J$+K$\IF LEN(J$)<7 THEN 510
540 L=L+1\Z=FND(0,L,J$)\J$=""
550 IF N<=K THEN 510
560 FOR I=1TO800\NEXT\ FORY=47TO30STEP-3\ FORX=0TO12STEP2
570 PLOT X,Y,0\ NEXT\NEXT\ GOTO 240
580 REM: DICE
590 Z=FNA(3,8,63)\ FOR N=1 TO RND(0)*15+4
600 D1=INT(RND(0)*6)+1 \ D2=INT(RND(0)*6)+1
610 Z=FNA(1,7,D1+176)\ Z=FNA(5,7,D2+176)\ NEXT
620 Z=FNF(1,7,5)\ Z=FNF(5,7,5)\ D3=0
630 IF D1=D2 THEN D3=1\ Z=FNA(3,8,170)\ Z=FNF(3,8,5)
640 RETURN
650 RESTORE 710\ FOR V=-4 TO 4\ IF V=0 THEN 700
>

```


LIST 651,1200

```
660 READ T,N\G=SGN(V)\B(T)=G*N
670 FOR X=1 TO N\ W=FNX(T)\W2=FNX(T,X)
680 A=FNS(T,G)\B=FNS(T,G*2)
690 Z=FNA(W,W2,A)\Z=FNA(W+1,W2,B)
700 NEXT\NEXT\RETURN
710 DATA 6,5,8,3,13,5,24,2,1,2,12,5,17,3,19,5
720 RESTORE 730\ FORI=1TO16\READ R(I)\NEXT\RETURN
730 DATA 60,52,38,39,60,44,37,39
740 DATA 57,41,13,15,57,25,11,15
750 REM: PLT BD\ !CHR$(12)\ F=15
760 !CHR$(12)\ PLOT 0,43,0\ F=15
770 FOR X=15 TO 112\ PLOT X,5,1\ PLOT X,42,1\ NEXT
780 A=19\ FOR X=F TO F+48 STEP 8\ FOR Y=6 TO 41\ P=0
790 IF X=F OR X=F+48 OR Y<A OR Y>48-A THEN P=1
800 PLOT X,Y,P\ NEXT\ IF A=19 THEN A=18 ELSE A=19\ NEXT
810 F=F+49 \ IF F<=64 THEN 780
820 FOR P=1 TO 24\ X=FNX(P)\ Y=FNX(P,-1)\ Z=FNA(X,Y,P+192)
830 NEXT\RETURN
840 DEF FNA(X,Y,V) \ REM: PUT V @ ADDR X,Y; RET OLD V.
850 A1=Y*64+X+63488\A2=PEEK(A1)
860 POKE A1,V \ RETURN A2
870 FNEND
880 DEF FNX(P) \ REM: RET X ADDR OF POS P.
890 Z=P \ IF P>12 THEN Z=25-P
900 RETURN 57-(Z*4)
910 FNEND
920 DEF FNY(P,N) \ REM: RET Y ADDR OF POS(P)+N.
930 IF N>8 THEN RETURN -2
940 Y1=1+N \ IF P>12 THEN Y1=14-N
950 RETURN Y1
960 FNEND
970 DEF FNQ(P) \ REM: +1 IF B(P) LEFT HALF, ELSE 0.
980 Q1=P-1\ Q1=(INT(Q1/6)+1)/4\ Q1=Q1-INT(Q1)\ Q1=INT(Q1*2)
990 RETURN Q1
1000 FNEND
1010 DEF FNS(P,X)
1020 S1=0 \ IF X<0 THEN S1=4
1030 S1=S1+(FNQ(P)*2)+ABS(X) \ IF P>12 THEN S1=S1+8
1040 RETURN R(S1)
1050 FNEND
1060 DEF FNM(F,T)
1070 M1=ABS(B(F))
1080 IF F>24 THEN 1100
1090 M2=FNX(F)\ M3=FNY(F,M1)\ M8=FNX(M2,M3,6)
1100 M1=B(F) \ M2=SGN(M1) \ B(F)=M1-M2
1110 IF T=0 THEN RETURN 0
1120 M1=B(T)\ B(T)=M1+M2\ M1=ABS(B(T))\ M3=FNX(T)
1130 M4=FNY(T,M1)\ M5=FNS(T,M2*1)\ M6=FNS(T,M2*2)
1140 M7=FNA(M3,M4,M5)\ M7=FNA(M3+1,M4,M6)\ M8=FNX(M3,M4,5)
1150 RETURN 0
1160 FNEND
1170 DEF FNF(X,Y,T)
1180 F1=FNA(X,Y,63)\ F3=A1\ F2=FNA(X+1,Y,63)\ F4=A1
1190 FOR F5=1 TO T \ FOR F6=1 TO 35 \ NEXT
1200 F7=PEEK(F3) \ F8=PEEK(F4) \ POKE F3,F1 \ POKE F4,F2
```

>


```

LIST 1201,1750
1210 F1=F7 \ F2=F8
1220 NEXT \ RETURN 0
1230 FNEND
1240 DEF FNG(T) \REM: INPUT MOVE
1250 IF INP(0)>0 THEN 1270
1260 G8=(T-1)*4+1\ G1=FNA(G8,8,156)\ G1=FND(0,3,W$+"S")
1270 G1=FND(0,4,"MOVE:")
1280 IF INP(0)=0 THEN 1280 ELSE G2=INP(1)
1290 IF G2=13 THEN G2=-37\ G3=-1\ GOTO 1330
1300 IF G2=32 THEN G2=-32\RETURN -2
1310 IF G2=27 THEN G2=-19\G3=0\GOTO 1330
1320 G2=G2-64\ G3=G2\ IF G2<0 OR G2>24 THEN 1250
1330 G1=FNA(6,4,G2+192)\ G1=FND(0,3,"")
1340 G1=FND(0,4,"")\ G1=FNA(G8,8,63)\ RETURN G3
1350 FNEND
1360 DEF FNI$(M$,I,X,Y)\ !CHR$(12)
1370 I$=""\I4=LEN(M$)/2+1\ I1=FND(32-I4,2,M$)\I1=1
1380 IF INP(0)=0 THEN 1380 ELSE I2=INP(1)
1390 IF I2=13 THEN 1420 ELSE IF I1>I THEN 1380
1400 I$=I$+CHR$(I2)\ I3=FNA(X-1+I1,Y,I2+128)
1410 I1=I1+1\ GOTO 1380
1420 RETURN I$
1430 FNEND
1440 DEF FND(X,Y,Z$) \ REM: DISPLAY Z$ @X,Y
1450 FOR D5=1 TO LEN(Z$)
1460 D6=ASC(Z$(D5,D5)) \ D7=FNA(X-1+D5,Y,D6+128)
1470 NEXT \ RETURN 0
1480 FNEND
1490 DEF FNB(P) \ REM: RET 1 IF BEAR OFF OK
1500 B1=1\B2=18\B3=1\ IF P=-1 THEN B1=24\B2=7\B3=-1
1510 FOR B4=B1 TO B2 STEP B3\ B5=SGN(B(B4))
1520 IF B5=P THEN RETURN 0
1530 NEXT\ RETURN 1
1540 FNEND
1550 DEF FNH(H1,H2)
1560 Z5=((H1*H1+H1)/2)+25\Z6=B(Z5)
1570 IF H2>0 THEN B(Z5)=Z6+H1\Z6=B(Z5)
1580 Z1=ABS(Z6)\Z2=INT((Z1-1)/3)\Z3=Z1-(Z2*3)
1590 Z1=57+(Z3-1)*2 \ Z3=12-Z2 \ Z2=Z2+3
1600 IF H1<0 THEN Z2=Z3
1610 Z3=FNS(24,H1) \ Z4=FNS(24,H1*2)
1620 Z=FNA(Z1,Z2,Z3) \ Z=FNA(Z1+1,Z2,Z4)
1630 Z3=15 \ IF H2<1 THEN Z3=14
1640 Z=FNF(Z1,Z2,Z3)
1650 RETURN 0
1660 FNEND
1670 DEF FNW(P)\ FOR W1=1 TO 26
1680 IF SGN(B(W1))=P THEN EXIT 1710
1690 NEXT\!CHR$(12)\PLOT30,24,0
1700 !W$+" WON !!!"\GOTO 1750
1710 RETURN 0
1720 OUT 0,131
1730 IF INP(0)=0 THEN RETURN ELSE I=INP(1)
1740 IF I<128 THEN 1730 ELSE RETURN
1750 REM
>

```


PROGRAM NAME: HANGMAN

TYPE: GAME

HARDWARE REQUIREMENTS: 16K RAM

SOFTWARE REQUIREMENTS: BASIC VERSION A00

BRIEF DESCRIPTION:

You have wandered into Hangman Country and must use your skill to get out. The hangman will tell you the number of letters in a word. You must guess the correct letters with no more than 6 wrong guesses or hang.

OPERATING INSTRUCTIONS:

When the message "WELCOMETO THE HANGMAN!" appears, you can let the program time out (about 25 seconds) or type a carriage return to continue sooner. The game display will then appear -- gallows on the right. The number of letters in the word to be guessed will be represented by underlines. You can then type the letters you guess, one at a time (or all at once if you are brave enough). If your guess is correct, the underline or underlines corresponding to your guess will be replaced by the letter.

If your guess is incorrect, it will be displayed as a "used letter" and a body part will be drawn on the gallows. The first six incorrect guesses draw your body parts; the seventh hangs you.

You win (survive) if you can guess the entire word with less than seven mistakes.

You can change the word list used by modifying the program. The DIM statement (program line 1160) should dimension L\$ to 10 times the total number of words in the list. Line number 1170 should be a DATA statement with the total number of words. The list can be extended between

(but not including) line numbers 1170 and 1240 using the existing for-mat.

TYPE: GAME

HARDWARE REQUIREMENTS: 16K RAM

SOFTWARE REQUIREMENTS: BASIC VERSION 400

BRIEF DESCRIPTION:

You have wandered into Hangman Country and must use your skill to get out. The hangman will tell you the number of letters in a word. You must guess the correct letters with no more than 8 wrong guesses or hang.

OPERATING INSTRUCTIONS:

When the message "WELCOME TO THE HANGMAN!" appears, you can let the program time out (about 25 seconds) or type a carriage return to continue sooner. The game display will then appear -- gallows on the right. The number of letters in the word to be guessed will be represented by underlines. You can then type the letters you guess, one at a time (or all at once if you are brave enough). If your guess is correct, the underline or underlines corresponding to your guess will be replaced by the letter.

If your guess is incorrect, it will be displayed as a "used letter" and a body part will be drawn on the gallows. The first six incorrect guesses draw your body parts; the seventh hangs you.

You win (survive) if you can guess the entire word with less than seven mistakes.

You can change the word list used by modifying the program. The DIM statement (program line 1160) should dimension L\$ to 10 times the total number of words in the list. Line number 1170 should be a DATA statement with the total number of words. The list can be extended between


```

LIST 0,650
100 GOSUB 1150\ REM: SETUP WORD VECTOR
110 DIM M(7,13,2)
120 Z=RND(ABS(SIN(TIME(0))))
130 GOSUB 980
140 F2=0 \ REM: HANGMAN LOGIC
150 GOSUB 390\ GOSUB 600
160 C=0
170 K=LEN(W$)\G$=""\FOR I=1TOK\G$=G$+"-"+\NEXT
180 F1=1\ GOTO 250
190 A$=FNIS(0)\ F=0\F1=0
200 FOR I=1 TO K\IF W$(I,I)<>A$(1,1) THEN 220
210 F=1\G$(I,I)=A$(1,1)
220 IF G$(I,I)="-" THEN F1=1
230 NEXT
240 IF F=0 THEN GOSUB 350
250 PLOT 0,30,0\PRINTG$\PLOT 0,27,0
260 IF F1=1 THEN 190
270 PLOT 0,15,0\PRINT"CONGRATULATIONS!"
280 PRINT"THERE MAY BE SOME HOPE FOR YOU AFTER ALL!"
290 FOR I=1 TO 2500\NEXT\PRINTCHR$(12)\PLOT 0,43,0
300 PLOT 26,32,0\ INPUT "CARE TO TRY AGAIN, TURKEY? ",B$
310 B$=B$+" "\ IF B$(1,1)<>"N" THEN 150
320 PRINT\PRINT\PRINT" SO LONG CHICKEN! COME BACK NEXT TIME YOU"
330 PRINT" HAVE GALLOWS FEVER!"\PRINT\PRINTTAB(40),"- THE HANGMAN -"
340 GOTO 1240
350 U$=U$+A$\ C=C+1
360 PLOT 0,15,0\PRINT"USED LETTERS: ",U$\PLOT 0,12,0
370 Z=FNH(C)\ IF Z=0 THEN RETURN
380 GOTO 290
390 IF F2=1 THEN 560
400 PRINTCHR$(12)\PRINT\PRINT\PRINT\PRINTTAB(15),"WELCOME TO THE HANGMAN!"
410 PRINT\PRINT\PRINT
420 PRINT"THIS IS MY TERRITORY, AND I DON'T WELCOME INTRUDERS!"
430 PRINT\PRINT"IF YOU WISH TO LIVE TO SEE ANOTHER SUNSET, YOU MUST PROVE
440 PRINT"YOUR WORTH. IF YOU CAN GUESS MY WORD WITHOUT MAKING 7 MISTAKES"
450 PRINT"YOU MAY STAY. IF NOT, MY EXECUTIONS ARE SWIFT AND IMMEDIATE!"
460 PRINT\PRINTTAB(40),"SINCERELY,"\PRINTTAB(40),"-THE HANGMAN-"
470 FOR I=1 TO 1500
480 IF INP(0)<>0 THEN Z=INP(1)\ GOTO 500
490 NEXT
500 PRINTCHR$(12)\PLOT0,43,0\PLOT26,24,0\PRINT"YOU WILL BEGIN IMMEDIATELY
510 PLOT 0,21,0
520 RESTORE 1150\READ Q\ FORI=1TOQ
530 READ J$
540 IF LEN(J$)<10 THEN J$=J$+" "\ GOTO 540
550 L$=L$+J$\ NEXT
560 G$=""\ U$=""
570 A=FNH(0)
580 F2=1
590 RETURN
600 W=INT(RND(0)*314)\ W=W-(INT(W/Q)*Q)+1
610 W$=""
620 K=(W-1)*10\ FOR I=1TO10
630 IF L$(K+I,K+I)=" " THEN RETURN ELSE W$=W$+L$(K+I,K+I)
640 NEXT
650 RETURN
>

```



```

LIST 651,1200
660 DEF FNIS(X)
670 PLOT 0,24,0\PRINT"GUESS?"\PLOT 0,21,0
680 IF INP(0)=0 THEN 680 ELSE Z=INP(1)
690 IF Z>128 THEN Z=Z-128
700 IF Z<65 OR Z>91 THEN 680
710 PLOT 0,24,0\PRINT"          "\PLOT 0,21,0
720 RETURN CHR$(Z)
730 FNEND
740 DEF FNH(X)
750 IF X<>0 THEN GOTO 840
760 PRINTCHR$(12)\PLOT0,44,0
770 FOR I=1TO6\ PLOT 40,I,1\ NEXT
780 FOR I=40 TO 127\ PLOT I,7,1\ NEXT
790 FOR I=1 TO 47\PLOT 127,I,1\NEXT
800 FOR I=127 TO 50 STEP -1\PLOT I,47,1\NEXT
810 FOR I=47 TO 44 STEP -1\PLOT 50,I,1\NEXT
820 FOR I=47 TO 39 STEP -1\PLOT 83,I,1\ NEXT
830 RETURN X
840 IF X=7 THEN 890 ELSE I=1
850 IF M(X,I,1)=0 THEN 880
860 L=M(X,I,1)\V=M(X,I,2)\POKE L,V
870 I=I+1\ GOTO 850
880 RETURN 0
890 A1=63845\ W1=10\ L1=7
900 FOR I=47 TO 39 STEP -1\ PLOT 83,I,1\ NEXT
910 FOR Y=A1+64*L1 TO A1 STEP -64
920 FOR X=W1 TO 0 STEP -1
930 V=PEEK(Y+X)\ POKE Y+X+64,V
940 NEXT\ NEXT
950 PLOT 56,38,0\PRINT"YOU LOST!"\PLOT 0,35,0
960 RETURN -1
970 FNEND
980 N=0\FORI=1TO7\FORJ=1TO13\M(I,J,1)=0\M(I,J,2)=0\NEXT\NEXT
990 PRINTCHR$(12),"LOADING..."
1000 READ X\ IF X<-10 THEN RETURN
1010 IF X<0 THEN 1050
1020 IF X>100 THEN 1060
1030 N=N+1\M(K,N,1)=L\M(K,N,2)=X\L=L+1
1040 GOTO 1000
1050 K=-X\ N=0\ GOTO 1000
1060 L=X\ GOTO 1000
1070 DATA -1,63656,62,54,54,63719,62,7,47,47,6,63784,29,50,53
1080 DATA 31
1090 DATA -2,63849,48,55,63913,0,7,63977,0,7,64041,0,7
1100 DATA -3,63848,54,63910,62,43,59,63974,7,64037,62,15
1110 DATA -4,63850,54,55,63914,3,59,46,63981,7,64045,14
1120 DATA -5,64106,29,55,64171,59,46,64236,56,64300,56,54
1130 DATA -6,64104,53,31,64166,62,43,64230,56,64293,62,48
1140 DATA -999
1150 REM: HERE ON IS USER PROGRAMMABLE
1160 DIM LS(300)\RETURN
1170 DATA 30
1180 DATA "CONCISE","VERIFY","LITHOGRAPH","CONVERSION"
1190 DATA "AUTOMATIC","INTERIM","MISTAKE","LANTERN","GNU"
1200 DATA "XEROX","PAPER","REHEAT","SYSTEMS","PRODUCT"
>

```


LIST 1201,1240

1210 DATA"VERSION","DOCUMENT","DESCRIBE","CAPABLE"

1220 DATA"CHARACTER","RESTORE","NEXT","FISH","SLASH","TIGER"

1230 DATA"FLY","KNIGHT","LAB","LABORATORY","FOLLOW","PYREX"

1240 REM

>

PROGRAM NAME: WUMPUS

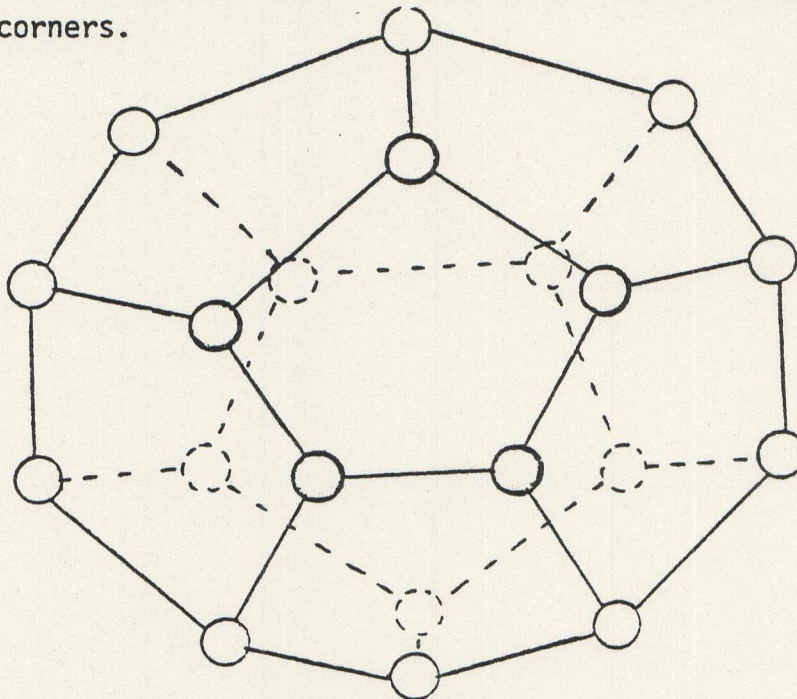
TYPE: GAME

HARDWARE REQUIREMENTS: 16K RAM

SOFTWARE REQUIREMENTS: BASIC VERSION A00

BRIEF DESCRIPTION:

You may not remember, but you have stumbled into the cave of the wumpus. The cave consists of 20 rooms; each is connected to three other rooms. To understand how the room positions relate, look at a dodecahedron (12 sided geometric shape with 20 corners). Each room corresponds to one of the twenty corners.



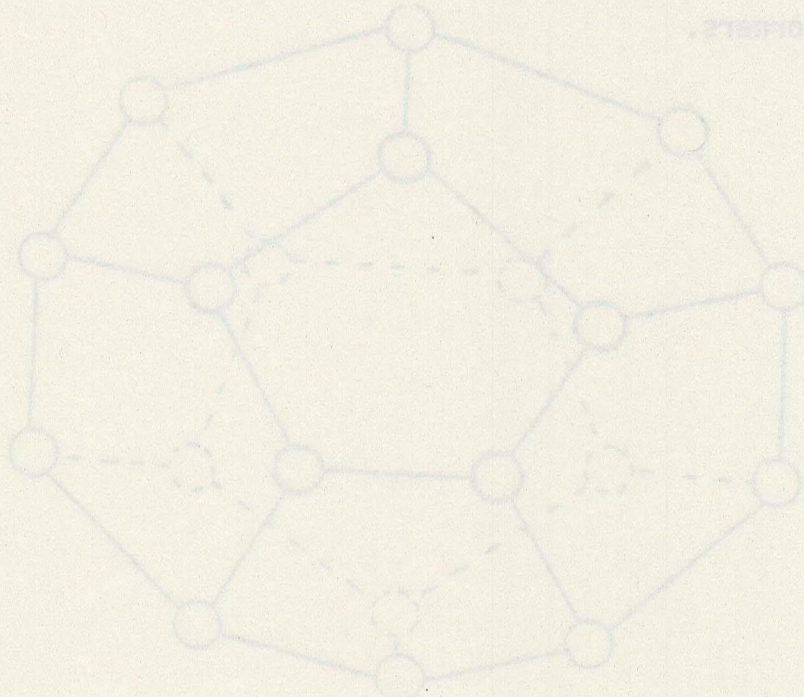
The object of the game is to remain alive long enough to shoot the wumpus. You have only five arrows.

OPERATING INSTRUCTIONS:

You will be given the three room numbers adjacent to your position. You will also be given warnings when you are adjacent to a hazard. By moving around you try to obtain enough information to shoot the wumpus. Complete instructions are given upon request when the game begins.

DESCRIPTION OF HOW PROGRAM WORKS:

Your starting position, and the starting positions of the Wumpus and hazards are randomly chosen by the computer. Your actions decide the intermediate steps and outcome.




```

LIST 0,650
100 REM *** WUMPUS ***
110 DIM P(5)
120 K=RND(TIME(1)/65536)
130 !CHR$(12),TAB(22),"HUNT THE WUMPUS"
140 INPUT"WOULD YOU LIKE INSTRUCTIONS? (Y OR N)",I$
150 IF I$="N" THEN 170 ELSE GOSUB 690
160 REM-SET UP CAVE (DODECAHEDRAL NODE LIST)
170 DIM S(20,3)
180 FOR J=1 TO 20
190 FOR K=1 TO 3
200 READ S(J,K)\NEXT\NEXT
210 DATA 2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
220 DATA 5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
230 DATA 10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
240 DATA 15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
250 DEF FNA(X)=INT(20*RND(0))+1
260 DEF FNB(X)=INT(3*RND(0))+1
270 DEF FNC(X)=INT(4*RND(0))+1
280 REM-LOCATE L ARRAY ITEMS
290 REM-1-YOU,2-WUMPUS,3&4-PITS,5&6-BATS
300 DIM L(6)\DIM M(6)
310 FOR J=1 TO 6
320 L(J)=FNA(0)
330 M(J)=L(J)\NEXT
340 REM-CHECK FOR CROSSOVERS (I.E. L(1)=L(2),ETC.)
350 FOR J=1 TO 6
360 FOR K=J TO 6
370 IF J=K THEN 390
380 IF L(J)=L(K) THEN 310
390 NEXT\NEXT
400 REM-SET # ARROWS
410 A=5
420 L=L(1)
430 REM-RUN THE GAME
440 ! "HUNT THE WUMPUS!"
450 REM-HAZARD WARNINGS & LOCATION
460 GOSUB 1160
470 REM-MOVE OR SHOOT
480 GOSUB 1310
490 ON O GOTO 510,550
500 REM-SHOOT
510 GOSUB 1370
520 IF F=0 THEN 480
530 GOTO 570
540 REM-MOVE
550 GOSUB 1650
560 IF F=0 THEN 460
570 IF F>0 THEN 620
580 REM-LOSE
590 ! "HA HA HA - YOU LOSE!"
600 GOTO 630
610 REM-WIN
620 ! "HEE HEE HEE - THE WUMPUS'LL GETCHA NEXT TIME!!"
630 FOR J=1 TO 6
640 L(J)=M(J)
650 NEXT J

```

>


```

LIST 651,1200
660 INPUT "SAME SET-UP (Y-N)",I$
670 IF I$<>"Y" THEN 310
680 GOTO 410
690 REM-INSTRUCTIONS
700 ! "WELCOME TO 'HUNT THE WUMPUS'"
710 ! "THE WUMPUS LIVES IN A CAVE OF 20 ROOMS. EACH"
720 ! "ROOM HAS 3 TUNNELS LEADING TO OTHER ROOMS. (LOOK"
730 ! "AT A DODECAHEDRON TO SEE HOW THIS WORKS-IF YOU"
740 ! "DON'T KNOW WHAT A DODECAHEDRON IS, ASK SOMEONE)"
750 !
760 ! "HAZARDS:"
770 ! "BOTTOMLESS PITS - TWO ROOMS HAVE BOTTOMLESS PITS",
780 ! "IN THEM."
790 ! "IF YOU GO THERE, YOU FALL INTO THE PIT (& LOSE!)"
800 ! "SUPER BATS - TWO OTHER ROOMS HAVE SUPER BATS. IF YOU"
810 ! "GO THERE, A BAT GRABS YOU AND TAKES YOU TO SOME",
820 ! "OTHER"
830 ! "ROOM AT RANDOM. (WHICH MIGHT BE TROUBLESOME)"
840 GOSUB 1920
850 ! "WUMPUS:"
860 ! "THE WUMPUS IS NOT BOTHERED BY THE HAZARDS (HE HAS",
870 ! "SUCKER"
880 ! "FEET AND IS TOO BIG FOR A BAT TO LIFT). USUALLY HE"
890 ! "IS ASLEEP. TWO THINGS WAKE HIM UP: YOUR ENTERING"
900 ! "HIS ROOM OR YOUR SHOOTING AN ARROW."
910 ! "IF THE WUMPUS WAKES, HE MOVES (P=.75) ONE ROOM"
920 ! "OR STAYS STILL (P=.25). AFTER THAT, IF HE IS WHERE YOU"
930 ! "ARE, HE EATS YOU UP (AND YOU LOSE!)."
940 !
950 ! "YOU:"
960 ! "EACH TURN YOU MAY MOVE OR SHOOT A CROOKED ARROW."
970 ! "MOVING: YOU CAN GO ONE ROOM (THRU ONE TUNNEL)"
980 GOSUB 1920
990 ! "ARROWS:"
1000 ! "YOU HAVE 5 ARROWS. YOU LOSE WHEN YOU RUN OUT. EACH"
1010 ! "ARROW CAN GO FROM 1 TO 5 ROOMS. YOU AIM BY TELLING"
1020 ! "THE COMPUTER THE ROOM #S YOU WANT THE ARROW TO GO TO."
1030 ! "IF THE ARROW CAN'T GO THAT WAY(IE NO TUNNEL) IT MOVES"
1040 ! "AT RANDOM TO THE NEXT ROOM."
1050 ! "IF THE ARROW HITS THE WUMPUS, YOU WIN."
1060 ! "IF THE ARROW HITS YOU, YOU LOSE."
1070 !
1080 ! "WARNINGS:"
1090 ! "WHEN YOU ARE ONE ROOM AWAY FROM WUMPUS OR HAZARD,"
1100 ! "THE COMPUTER SAYS:"
1110 ! "WUMPUS- 'I SMELL A WUMPUS'"
1120 ! "BAT - 'BATS NEARBY'"
1130 ! "PIT - 'I FEEL A DRAFT'"
1140 GOSUB 1920
1150 RETURN
1160 REM-PRINT LOCATION & HAZARD WARNINGS
1170 !
1180 FOR J=2 TO 6
1190 FOR K=1 TO 3
1200 IF S(L(1),K)<>L(J) THEN 1270
>

```



```

LIST 1201,1750
1210 ON J-1 GOTO 1220,1240,1240,1260,1260
1220 ! "I SMELL A WUMPUS!"
1230 GOTO 1270
1240 ! "I FEEL A DRAFT"
1250 GOTO 1270
1260 ! "BATS NEARBY!"
1270 NEXT\NEXT
1280 ! "YOU ARE IN ROOM ",L(1)
1290 ! "TUNNELS LEAD TO ",S(L,1),S(L,2),S(L,3)
1300 !\RETURN
1310 REM-CHOOSE OPTION
1320 INPUT "SHOOT OR MOVE (S-M)?",I$
1330 IF I$<>"S" THEN 1350
1340 O=1\RETURN
1350 IF I$<>"M" THEN 1320
1360 O=2\RETURN
1370 REM-ARROW ROUTINE
1380 F=0
1390 REM-PATH OF ARROW
1400 INPUT "# OF ROOMS (1-5)?",R
1410 IF ((R<1)OR(R>5)) THEN 1400
1420 FOR K=1 TO R
1430 INPUT "ROOM #?",P(K)
1440 NEXT
1450 REM-SHOOT ARROW
1460 A=A-1\L=L(1)\K=1
1470 FOR Q=1 TO 3
1480 IF S(L,Q)=P(K) THEN L=P(K)\GOTO 1520
1490 NEXT
1500 REM-NO TUNNEL FOR ARROW
1510 L=S(L,FNB(1))
1520 IF L<>L(2) THEN 1540
1530 !"HURRAY! YOU GOT THE WUMPUS!"\F=1\RETURN
1540 IF L=L(1) THEN !"OUCH! ARROW GOT YOU!"\GOTO 1590
1550 K=K+1\IF K<=R THEN 1470
1560 !"MISSED!"
1570 GOSUB 1600
1580 IF A>0 THEN RETURN ELSE !"OUT OF ARROWS!"
1590 F=-1\RETURN
1600 REM-MOVE WUMPUS ROUTINE
1610 K=FNC(0)\IF K=4 THEN 1640 ELSE L(2)=S(L(2),K)
1620 IF L(2)<>L THEN 1640
1630 ! "TSK TSK TSK - WUMPUS GOT YOU!"\F=1
1640 RETURN
1650 REM-MOVE ROUTINE
1660 F=0
1670 INPUT "WHERE TO? ",L
1680 IF (L<1) OR (L>20) THEN 1670
1690 FOR K=1 TO 3
1700 REM- CHECK IF LEGAL MOVE
1710 IF S(L(1),K)=L THEN 1750
1720 NEXT
1730 IF L=L(1) THEN 1750 ELSE !"NOT POSSIBLE-"\GOTO 1670
1740 REM-CHECK FOR HAZARDS
1750 L(1)=L

```

>


```

LIST 1751,1930
1760 REM-WUMPUS
1770 IF L<>L(2) THEN 1830
1780 ! "...OOPS! BUMPED A WUMPUS!"
1790 REM-MOVE WUMPUS
1800 GOSUB 1610
1810 IF F=0 THEN 1830 ELSE RETURN
1820 REM-PIT
1830 IF ((L<>L(3)) AND (L<>L(4))) THEN 1870
1840 ! "YYYYIIIIIEEEE...FELL IN PIT"
1850 F=-1\RETURN
1860 REM-BATS
1870 IF ((L<>L(5)) AND (L<>L(6))) THEN 1910
1880 ! "ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!"
1890 L=FNA(1)
1900 GOTO 1750
1910 RETURN
1920 INPUT "TYPE RETURN NOW TO CONTINUE.....",CS\!CHR$(12)
1930 RETURN
>

```


PROGRAM NAME: CRAPS

TYPE: GAME

HARDWARE REQUIREMENTS: 16K RAM

SOFTWARE REQUIREMENTS: BASIC VERSION A00

BRIEF DESCRIPTION:

Do you have gambling fever? If so, this is the game for you, the classic dice game of craps.

OPERATING INSTRUCTIONS:

Use the keyboard to enter your bet. The dice are automatically rolled and the results displayed. Full instructions are given upon request when the game begins.

DESCRIPTION OF HOW THE PROGRAM WORKS:

Random factors determine the numbers on the dice and the number of times the dice turn over. The dice are plotted accordingly. Results follow standard craps rules.


```

LIST 0,650
100 REM:*** CRAPS *** G.A. McComb, 03/29/77
110 DIM M$(64)
120 DIM E(13), F(13), A(7), M(6,7)
130 !CHR$(12)\!\!\!TAB(18),"WELCOME TO THE GAME OF CRAPS!"
140 !\!\!\!TAB(19),"DO YOU WANT INSTRUCTIONS?"\INPUT"",M$
150 M$=M$+" \" IF M$(1,1)="Y" THEN GOSUB 930
160 X=RND(TIME(1)/65536)
170 GOSUB 1260
180 Y=INT(250*X)+50
190 A=65
200 V=FNV(1,Y)
210 Z=1\ V=FNV(-2,0)
220 IF Z=1 THEN 260
230 V=FNV(1,Y)
240 IF Y=0 THEN 500
250 V=FNM(-1,"CHANGE THE DICE, TO THE LEFT...")
260 B=FNB(0)\ V=FNV(0,B)\ V=FNM(0,"")
270 IF B=0 THEN 820
280 IF ABS(B)<=Y THEN L=0\ GOTO 310
290 V=FNM(-1,"SORRY PAL, WE DON'T EXTEND CREDIT!")
300 GOTO 260
310 GOSUB 1120
320 L=L+1\ V=FNV(-6,L)
330 F=C+D
340 IF L>1 THEN 530
350 E=C+D
360 IF Z=2 THEN 630
370 ON E GOTO 410,410,410,380,380,380,450,380,380,380,450,410
380 V=FNM(1,"YOUR POINT IS "+STR$(E))
390 V=FNV(-4,E)
400 GOTO 850
410 V=FNM(1,"YOU CRAP OUT...")
420 Y=Y-B
430 IF Y=0 THEN 500
440 GOTO 490
450 V=FNM(1,"YOU PASS THIS TIME...")
460 GOTO 480
470 V=FNM(1,"CONGRATULATIONS! YOU MADE YOUR POINT.")
480 Y=Y+B
490 V=FNV(1,Y)\ IF Y>0 THEN 260
500 V=FNV(-1,0)\ V=FNV(1,0)
510 V=FNM(1,"YOU HAVE GONE BANKRUPT... SORRY!")
520 GOTO 860
530 IF Z=2 THEN 710
540 IF F=E THEN 470
550 IF F=7 THEN 580
560 V=FNM(-1,"... ROLL AGAIN.")
570 GOTO 310
580 V=FNM(1,"YOU LOSE!")
590 Z=2\ V=FNV(-3,0)
600 Y=Y-B
610 IF Y=0 THEN 500
620 GOTO 230
630 ON E GOTO 670,670,670,640,640,640,690,640,640,640,690,670
640 V=FNM(1,"MY POINT IS "+STR$(E))
650 V=FNV(-4,E)
>

```



```

LIST 651,1200
660 GOTO 850
670 V=FNM(1,"I CRAP OUT...")
680 GOTO 480
690 V=FNM(1,"I PASS...")
700 GOTO 420
710 IF F=E THEN 800
720 IF F=7 THEN 750
730 V=FNM(1,"I ROLL AGAIN...")
740 GOTO 310
750 V=FNM(1,"I LOSE... DAMN!")
760 Y=Y+B
770 IF Y=0 THEN 500
780 Z=1\ V=FNV(-2,0)
790 GOTO 230
800 V=FNM(1,"I MADE MY POINT!")
810 GOTO 420
820 !CHR$(12)\!\!\!\!"SO LONG CHICKEN!"
830 !\!\!"You're lucky you quit while you were ahead..."
840 GOTO 1980
850 GOTO 310
860 !CHR$(12)\ PLOT 0,43,0
870 A=65
880 V=FNM(1,"DO YOU WANT TO PLAY ANOTHER GAME?")
890 PLOT 100,6,0\ INPUT "",M$\ PLOT 0,5,0
900 M$=M$+" "\ IF M$(1,1)="Y" THEN 160
910 !CHR$(12)
920 GOTO 1980
930 !CHR$(12)
940 !"CRAPS IS A GAME WHERE YOU BET ON DICE ROLLS."
950 !\!"The first roll is your point roll. If you get a 2, 3,"
960 !"or 12 on your point roll, you lose. If you get a 7 or 11"
970 !"you win. Any other roll is your 'Point'. On subsequent"
980 !"rolls you must match your point to win. If you get a 7"
990 !"before matching your point you lose. That's all there"
1000 !"is to it!"
1010 !\!\!\INPUT "HIT RETURN TO CONTINUE...",M$
1020 !CHR$(12)
1030 !\!"To bet, you decide to bet for yourself (that you will"
1040 !"win), or against yourself (that you will lose). You bet"
1050 !"for yourself with a positive bet, against with a negative"
1060 !"bet. When the computer has the dice, you bet for the "
1070 !"computer with a negative bet (you win if the computer"
1080 !"wins), against the computer with a positive bet (you lose"
1090 !"if the computer wins)."
```

1100 !\!"If you need help betting, type 'B' when asked to bet."

```

1110 !\!\INPUT "HIT RETURN TO CONTINUE...",M$\ RETURN
1120 REM: ROLL DICE
1130 IF Z=2 THEN 1180
1140 IF INP(0)<>0 THEN 1170
1150 IF A=65 THEN 1180
1160 V=FNM(1,"HIT RETURN TO ROLL DICE")
1170 IF INP(0)=0 THEN 1170 ELSE A=INP(1)
1180 V=FNM(0,"")
1190 J=RND(0)*15+5
1200 FOR I=1 TO J
>

```



```

LIST 1201,1750
1210 C=INT(RND(0)*6)+1\ D=INT(RND(0)*6)+1
1220 FOR K=1 TO 7
1230 POKE A(K),M(C,K)\ POKE A(K)+27,M(D,K)
1240 NEXT\ NEXT
1250 RETURN
1260 REM: LOAD DICE AND PLOT DICE CUBES
1270 RESTORE 1910
1280 FOR I=1 TO 7\ READ A(I)\ NEXT
1290 FOR I=1 TO 6\ FOR J=1 TO 7
1300 READ M(I,J)\ NEXT\ NEXT
1310 !CHR$(12)\ PLOT 0,43,0
1320 FOR I=22 TO 51\ PLOT I,29,1\ PLOT I+54,29,1\ NEXT
1330 FOR I=30 TO 43\ PLOT 50,I,1\ PLOT 51,I,1
1340 PLOT 104,I,1\ PLOT 105,I,1\ NEXT
1350 FOR I=49 TO 22 STEP -1\ PLOT I,43,1\ PLOT I+54,43,1\ NEXT
1360 FOR I=42 TO 30 STEP -1\ PLOT 22,I,1\ PLOT 23,I,1
1370 PLOT 76,I,1\ PLOT 77,I,1\ NEXT
1380 RETURN
1390 DEF FNV(X,N)
1400 IF X>0 THEN 1420
1410 ON ABS(X)+1 GOTO 1440,1460,1480,1500,1520,1540,1570
1420 PLOT 16,18,0\ !"Your account: ",N," "\ PLOT 0,15,0
1430 RETURN 0
1440 PLOT 16,15,0\ !"Your bet: ",N," "\ PLOT 0,12,0
1450 RETURN 0
1460 PLOT 16,15,0\ !" "\ PLOT 0,12,0
1470 RETURN 0
1480 PLOT 56,47,0\ !"YOUR DICE"\ PLOT 0,43,0
1490 RETURN 0
1500 PLOT 56,47,0\ !" MY DICE "\ PLOT 0,43,0
1510 RETURN 0
1520 PLOT 92,15,0\ !"Point: ",N\ PLOT 0,12,0
1530 RETURN 0
1540 PLOT 92,18,0\ !" "\ PLOT 0,15,0
1550 PLOT 92,15,0\ !" "\ PLOT 0,12,0
1560 RETURN 0
1570 PLOT 92,18,0\ !"Rolls: ",N\ PLOT 0,15,0
1580 RETURN 0
1590 FNEND
1600 DEF FNM(X,M$)
1610 IF X<>0 THEN 1650
1620 REM: CLEAR MESSAGE
1630 FOR I=0 TO 63\ POKE 64320+I,63\ NEXT
1640 S=0\ RETURN 0
1650 IF S>1 THEN FOR I=0 TO 63\ POKE 64320+I,63\ NEXT
1660 S=LEN(M$)\ IF S=0 THEN RETURN 0
1670 M=65-S\ PLOT M,6,0\ !M$\ PLOT 0,5,0
1680 FOR I=1 TO 500+(5*S)\ NEXT I
1690 RETURN 0
1700 FNEND
1710 DEF FNB(X)
1720 V=FNv(-5,0)\ V=FNv(-1,0)
1730 V=FNM(1,"How much do you want to bet?")
1740 PLOT 94,6,0\ INPUT "",M$\ PLOT 0,5,0
1750 N=LEN(M$)\ IF N=0 THEN 1730

```



```
LIST 1751,1980
1760 FOR I=1 TO N\ M=ASC(M$(I))
1770 IF M=46 THEN 1820
1780 IF M=45 THEN 1820
1790 IF M=36 THEN M$(I,I)="0"\ GOTO 1820
1800 IF M=66 THEN EXIT 1860
1810 IF M<48 OR M>57 THEN EXIT 1840
1820 NEXT
1830 RETURN VAL(M$)
1840 V=FNM...,"Bad input - Retype.")
1850 GOTO 1730
1860 GOSUB 1020
1870 GOSUB 1260
1880 V=FNV(1,Y)
1890 GOTO 1730
1900 FNEND
1910 DATA 63822,63694,63630,63698,63830,63702,63638
1920 DATA 63,63,63,54,63,63,63
1930 DATA 27,63,63,63,63,63,45
1940 DATA 27,63,63,54,63,63,45
1950 DATA 27,63,45,63,27,63,45
1960 DATA 27,63,45,54,27,63,45
1970 DATA 27,54,45,63,27,54,45
1980 REM
>
```